IN THE CLAIMS

Listing of Claims:

1. (currently amended) A method for managing a link stack comprising the steps of:

~~for a first type instruction:~~

setting a first data value, corresponding to a portion of a first address stored in said link stack, [[in]] into a first portion of an entry in a queue having a plurality of entries in response to a first type operation; [[and]]

~~for a second type instruction:~~

setting a current value of a first pointer ~~into~~ of said link stack [[in]] into a first register in response to a second type operation; and

setting said current value of said first pointer [[in]]into a second portion of said entry in said queue.

2. (currently amended) The method of claim 1 further comprising the steps of:

~~for said second type instruction,~~ reading from said link stack a second address stored at a stack entry at said current value of said first pointer in response to said second type operation; and

storing said second address in a second register.

3. (currently amended) The method of claim 1 wherein said first type ~~instruction~~ operation is a "push" type ~~instruction~~ operation [[an]] and said second type ~~instruction~~ operation is a "pop" type ~~instruction~~ operation.

4. (currently amended) The method of claim [[1]] 2 further comprising the step of ~~, for a first type operation,~~ setting said first data value in a third register for said first type operation.

5. (currently amended) The method of claim 4 further comprising the step of ~~, for third type instruction,~~ setting a second data value from said second register in a said first portion of said entry in said queue in response to a third type operation.

6. (currently amended) The method of claim 1 wherein said steps recited therein are performed in response to a fetch of an instruction using a corresponding one of said first type ~~instruction~~ operation and said second type ~~instruction~~ operation.

7. (currently amended) The method of claim 2 further comprising the steps of:

receiving a pointer value in response to a pipeline flush operation, said pointer value operable for pointing into said queue; and

setting a current pointer value of said first pointer to a value in said second portion of an entry in the queue pointed to by said pointer value.

8. (currently amended) The method of claim 7 further comprising the steps of:

retrieving a value from said second register; and

comparing said value from said retrieving step with a value from said second portion of said entry in the queue pointed to by said pointer value.

9. (currently amended) The method of claim 8 further comprising the steps of:

in response to a compare match in said comparing step, comparing said current value of said first pointer with a data value in said first register; and

in response to a ~~compare~~ match of said current value of said first pointer and said data value in said first register, setting said data value in said second register [[in]] into said link stack at a location pointed to by said current value of said first pointer decremented by one.

Claims 10-17 (canceled)

18. (currently amended) A data processing system comprising:

a central processing unit (CPU), said CPU including:

a link stack; and

first logic operable for ~~, for a first type instruction,~~ setting a first data value corresponding to a portion of a first address stored in said link stack [[in]] into a first

portion of an entry in a queue having a plurality of entries in response to a first type operation, and ~~for a second type instruction,~~ setting a current value of a first pointer into said link stack in a first register in response to a second type operation, and setting said current value of said first pointer in a second portion of said entry in said queue.

19 (currently amended) The system of claim 18 wherein said CPU further comprises, second logic operable for, for said second type ~~instruction~~ operation, reading from said link stack a second address stored at a stack entry at said current value of said first pointer, and storing said second address in a second register.

20. (currently amended) The system of claim 18 wherein said first type instruction is a "push" type ~~instruction~~ operation [[an]] and said second type ~~instruction~~ operation is a "pop" type ~~instruction~~ operation.

21. (currently amended) The system of claim [[18]] 19 wherein said CPU further comprises, third logic operable for~~, for a first type operation,~~ setting said first data value in a third register for a first type operation.

22. (currently amended) The system of claim 21 wherein said CPU further comprises fourth logic operable for, for third type ~~instruction~~ operation, setting a second data value from said second register in a said first portion of said entry.

23. (currently amended) The system of claim 18 wherein said first logic sets said first data value, for said first type instruction, and sets said current value of said pointer, for said second type ~~instruction~~ operation in response to a fetch of a corresponding one of said first type ~~instruction~~ operation and said second type ~~instruction~~ operation.

24. (currently amended) The system of claim 19 wherein said CPU further comprises:

fifth logic operable for receiving a pointer value in response to a pipeline flush operation, said pointer value operable for pointing into said queue; and

sixth logic operable for setting a current pointer value of said first pointer to a value in said second portion of an entry in the queue pointed to by said pointer value.

25. (currently amended) The system of claim 24 wherein said CPU further comprises:

seventh logic operable for retrieving a data value from said second register; and

eighth logic operable for comparing said value from said retrieving step with a data value from said second portion of said entry in the queue pointed to by said pointer value.

26. (currently amended) The system of claim 25 wherein said CPU further comprises:

ninth logic operable for, in response to a ~~compare~~ match in said comparing step, comparing said current value of said first pointer with a data value in said first register; and

tenth logic operable for, in response to a ~~compare~~ match of said current value of said first pointer and said data value in said first register, setting said data value in said second register [[in]]into said link stack at a location pointed to by said current value of said first pointer decremented by one.

27. (currently amended) The system of claim 18 further comprising system memory coupled to said CPU, said system memory operable for storing a program of instructions including ~~instructions of~~ said first type operations and said second type operations.

Claims 28-48 (canceled)